

E-FILED on 8/8/2013

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE DIVISION

NAZOMI COMMUNICATIONS, INC.,

No. C-10-05545 RMW

Plaintiff,

v.

**ORDER CONSTRUING CLAIMS OF U.S.
PATENT NO. 6,338,160**SAMSUNG TELECOMMUNICATIONS,
INC., *et al.*,

Defendants.

On November 28, 2012, the court held a claim construction hearing for the purpose of construing disputed terms in the claims of U.S. Patent No. 6,338,160 ("160 patent"), a continuation-in-part of U.S. Patent No. 6,332,215 ("215 parent patent"). The parties identified four terms in the '160 patent as the "most significant to the resolution of the case," and four additional disputed terms. Joint Claim Construction Statement ("JCCS"), Dkt. No. 232.¹ On October 19, 2012, defendants filed modified proposed claim constructions for three of the disputed terms. Dkt. No. 238.² The court

¹ Originally, the parties designated five terms as "most significant"; however, the term "jump table" is no longer at issue because Nazomi is no longer asserting dependent claim 12 of the '160 patent. *See* Pl.'s Opening Claim Construction Br. 4 n.1, Dkt. No. 239.

² Although the court does not generally condone the revision of a JCCS position after the deadline for filing the JCCS, here, Nazomi was put on fair notice of the revision and the revision does not materially alter the issues before the court. Accordingly, the court will consider defendants' revised

construes the "most significant terms" as identified by the parties. *See* Patent L.R. 4-3 (limiting parties to 10 most significant terms). After consideration of the claims, specification, prosecution history, and other relevant evidence, and after hearing the arguments of the parties, the court construes the disputed terms of the '160 patent as set forth below.

I. THE '160 PATENT

On January 8, 2002, the United States Patent and Trademark Office issued the '160 patent to inventors Mukesh K. Patel and Chitrabhanu Dasgupta, who assigned the patent to plaintiff Nazomi Communications, Inc. ("Nazomi"). The '160 patent, titled "Constant Pool Reference Resolution Method," is directed generally to a method of running Java programming language in software wherein "references to the constant pool are implemented by using a Data Resolution Field within the constant pool entry." '160 patent, Abstract. A "constant pool" is, generally, a data structure attached to a loaded Java class file. As the name implies, it contains constants that "encode all the names (of variables, methods, and so forth) used by any method in the class." *Id.* col.7 ll.43-47. In contrast to most computer languages, where all references to objects are "resolved" (by loading the objects into memory) prior to running the program, Java can resolve references to objects at runtime. *See id.* col.1 ll.44-48. In Java, an "invoke instruction," which references the runtime constant pool, "must cause the [unresolved] reference to be resolved." *Id.* col.1 ll.54-59. In the prior art, it often took "a considerable amount of time to resolve a reference each time that [an] instruction[, such as an invoke instruction, wa]s run." *Id.*

The '160 patent is directed to an improved method of resolving references to objects at runtime. The claimed constant pool includes a "data resolution field" (also called a "resolution data field") that indicates whether the reference to an object has been resolved. *Id.* col.2 ll.19-39. For example, a "0" in the data resolution field would indicate that the reference has not been resolved and "cause[] the system to jump to a location . . . in the native instruction region [where] the resolve instructions" are located; whereas a "1" in the field would indicate that the reference is resolved and cause the system to jump to the "native instruction for the invoke instruction." *Id.* col.8 ll.32-52; *see also id.* figs.9A and 9B. The invention also includes an "indication field," which, before resolution,

proposed constructions in this Order.

contains an "indication of a reference that may need resolution." *Id.* col.8 ll.32-52, col.10 ll.3-4, and fig.9A. Once the reference is resolved, the indication field is updated to "indicate[]" the location of the loaded object." *Id.* col.8 ll.40-48 and fig.9B.

II. LEGAL STANDARD

Claim construction is exclusively within the province of the court. *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 387 (1996). "It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.'" *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (internal citation omitted). Claim terms "are generally given their ordinary and customary meaning," defined as "the meaning . . . the term would have to a person of ordinary skill in the art in question . . . as of the effective filing date of the patent application." *Id.* at 1313 (internal citation omitted). The skilled artisan reads the claim term "in the context of the entire patent . . . including the specification." *Id.*; see also *Multiform Desiccants, Inc. v. Medzam, Ltd.*, 133 F.3d 1473, 1477 (Fed. Cir. 1998). In determining the meaning of a disputed claim limitation, the intrinsic evidence, including the claim language, written description, and prosecution history, is the most significant. *Phillips*, 415 F.3d at 1315-17. The court reads the claims in light of the specification, which is "the single best guide to the meaning of a disputed term." *Id.* at 1315. Furthermore, "the interpretation to be given a term can only be determined and confirmed with a full understanding of what the inventors actually invented and intended to envelop with the claim." *Id.* at 1316 (quoting *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998)). The words of the claims must be understood as the inventor used them as revealed by the patent and prosecution history. *Id.*

Although extrinsic evidence is less significant than the intrinsic record, the court may also reference extrinsic evidence to "shed useful light on the relevant art." *Id.* at 1317 (quoting *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 862 (Fed. Cir. 2004)). "[T]echnical dictionaries may provide [help] to a court 'to better understand the underlying technology' and the way in which one of skill in the art might use the claim terms. . . . Such evidence . . . may be considered if the court deems it helpful in determining 'the true meaning of language used in the patent claims.'" *Id.* at 1318 (internal citations omitted).

Nazomi asserts, and defendants do not dispute, that for the purposes of these patents, a person of ordinary skill in the art at the time of the invention would have either: (1) a bachelor of science in computer engineering, electrical engineering, or computer science with about three years of work experience in the field of computer architecture; or (2) a master's of science in computer engineering, electrical engineering, or computer science with about a year of work experience in the field of computer architecture. Pl.'s Opening Claim Construction Br. 1, Dkt. No. 239 ("Pl.'s Br.")

III. CLAIM CONSTRUCTION

A. "constant pool" / ("constant pool entry" / "entry in a constant pool")

Nazomi's Proposed Construction	Defendants' Proposed Construction
" constant pool ": a data structure that includes at least one constant pool entry	" constant pool ": a table attached to a single loaded class that encodes all the names that can be used by any method in the loaded class
" constant pool entry " / " entry in a constant pool ": an entry in the constant pool that includes data indicating that a reference requires resolution or that the reference is resolved	" constant pool entry " / " entry in a constant pool ": an entry within the constant pool

The parties primarily dispute whether the constant pool (i.e., the "data structure" or "table") must be "attached to a *single* loaded class." Nazomi Br. 9. Nazomi asserts that defendants' proposed construction would improperly import limitations from the '160 patent specification into the claims and improperly limit the claims to a specific implementation of the Java Virtual Machine ("JVM") that is disclosed in the JVM Specification (a specification documenting the design of the JVM). *See Phillips*, 415 F.3d at 1323 ("[W]e have repeatedly warned against confining the claims to . . . embodiments" in the specification.). Nazomi further argues that because the '160 patent specification uses open-ended language, the term is not limited to Java. *See* '160 patent col.7 ll.35-38 ("The term Java in the specification or claims should be construed to cover successor programming languages or other programming languages . . .").

Defendants counter that a person of ordinary skill in the art at the time of the invention would have understood the term "constant pool" to be a term of art specific to Java. Defendants, thus, direct the court to the JVM Specification as the single best example of how one of ordinary skill in the art at the time would have understood the term "constant pool." *See* JVM Specification

§ 3.5.5 (1st ed. 1997), Dkt. No. 245-31; JVM Specification § 3.5.5 (2d ed. 1999), Dkt. No. 245-17 (describing a runtime constant pool as a data structure attached to a single loaded class). According to defendants, the disclosures in the '160 patent specification comport with the JVM Specification, and confirm that the "constant pool" claimed is a data structure attached to a single loaded class. Defendants also advance a claim differentiation argument. Claim 1 requires "an instruction that references a *data structure*, the *data structure* storing an indication of a reference that may need resolution" Thus, according to defendants, "'constant pool entry' [in claim 11] must be something more specific than [the] generic 'data structure' [in claim 1]." Defs.' Responsive Br. 10, Dkt. No. 247(emphasis in brief).

The inquiry into how a person of ordinary skill in the art would understand a claim term is an "objective baseline from which to begin claim interpretation." *Phillips*, 415 F.3d at 1313. At the time of invention, the term "constant pool" and "constant pool entry" were terms of art particular to Java. *See generally* Decl. of Mark E. Levitt, Ph.D., "Java Programming Overview" and "Java Virtual Machine" headings, ¶¶ 1-8, Dkt. No. 251; Decl. of David I. August, Ph.D. ("August Decl.") ¶¶ 6-21, Dkt. No. 246-35; '160 patent; '215 parent patent (all consistently describing the invention in terms of Java). Nazomi asks the court to construe the term "constant pool" based *only* on intrinsic evidence, and not to "limit" the term based on embodiments in the specification. The problem with Nazomi's argument is that the term "constant pool" is a term of art without a plain and ordinary meaning, and thus the court must look somewhere to construe it.

The '160 patent never actually defines the term "constant pool." The specification first references "*the* constant pool" in the "background of the invention section" as if the reader is already familiar with the term. '160 patent col.1 l.58 (emphasis added). Later on in the specification, the term "constant pool" is described in the context of figure 8. *Id.* col.7 ll.41-47. In describing figure 8, the specification states: "*Every currently loaded class has a constant pool attached to it. The constant pool is allocated when a class is first loaded, the constants in this pool encode all the names (of variables, methods, and so forth) used by any method in the class.*" '160 patent col.7 ll.43-47 (emphases added). Because an embodiment in the specification is not generally limiting, and because there is no other "plain and ordinary meaning" disclosed in the specification, the court

"looks to 'those sources available to the public that show what a person of [ordinary] skill . . . would have understood disputed claim language to mean.'" *Phillips*, 415 F.3d at 1314 (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1116 (Fed. Cir. 2004)).

The court finds that the JVM Specification is exactly the type of instructive evidence that the Federal Circuit described as appropriate to consider in *Phillips*. While the court recognizes that extrinsic sources are often less reliable, *see id.* at 1318-19, Nazomi has failed to point the court to any extrinsic source for a definition of "constant pool" that is different than the definition provided in the JVM Specification. Because the '160 patent specification actually references the JVM Specification, and because Nazomi has not convinced the court that one of ordinary skill would have understood this term of art outside of the Java context at that time, the court finds the JVM Specification to be particularly reliable and instructive as to the meaning of the term "constant pool" to the ordinarily skilled artisan in this instance. The relevant editions of the JVM Specification define the terms "constant pool" and "runtime constant pool"³ as:

[A] *per-class* or *per-interface* runtime representation of the `constant_pool` table in a [Java] class file It contains several kinds of constants, ranging from numeric literals known at compile time to method and field references that must be resolved at run time. The runtime constant pool serves a function similar to that of a symbol table for a conventional programming language, although it contains a wider range of data than a typical symbol table. . . . The runtime constant pool for a class or interface is [constructed/created] when [a Java class file for the] class or interface is [created/loaded]

JVM Specification § 3.5.5 (1st ed. 1997)("constant pool"); JVM Specification § 3.5.5 (2d ed. 1999) ("runtime constant pool"). The description of the constant pool in the '160 patent specification is consistent with the JVM Specification because it likewise explains that "[e]very currently loaded class has a constant pool attached to it," and that "the *constant pool is allocated* when a class is first loaded." '160 patent col.7 ll.43-47 (emphases added). The specification confirms that the "constant pool" claimed *is* the Java-type constant pool described in the JVM Specification.

The court also agrees with defendants that the term "constant pool" in claim 11 is more specific than the term "data structure" in claim 1 under the doctrine of claim differentiation. *See*,

³ The 1st edition labels the relevant section as "constant pool," whereas the 2nd edition labels the relevant section as "runtime constant pool." However, both JVM Specifications use the exact same language, except for minor editorial differences indicated in brackets above, and both effectively define a "runtime constant pool."

e.g., *Seachange Int'l Inc. v. C-COR Inc.*, 413 F.3d 1361, 1368-69 (Fed. Cir. 2005) ("[T]here is still a presumption that two independent claims have different scope when different words or phrases are used in those claims."). In contrast to the generic term "data structure" in claim 1, the term "constant pool" in claim 11 is a term of art that connotes a more specific affiliation with a single loaded Java ".class" file. Nazomi fails to rebut the presumption that these claims are different in scope. *See* Pl.'s Reply Br. 9-10, Dkt. No. 257 (simply rehashing the argument that the specification "does not expressly define the term 'constant pool'" but rather "describes the constant pool as any type of data structure").

Because the defendants' proposed construction is consistent with the only disclosure in the specification and the extrinsic evidence before the court, the court adopts the defendants' proposed construction, replacing the word "table" with "data structure." In the original JCCS, defendants' proposed construction used the term "data structure" instead of "table." Defendants offer no substantial explanation for the change to the word "table," and this court finds no reason to limit the data structure to a table. Therefore, the court construes the term "constant pool" to mean: "a data structure attached to a single loaded class that encodes all the names that can be used by any method in the loaded class." Likewise, the court construes the term "constant pool entry" to mean: "an entry within the constant pool [as defined *supra*]."

B. "instruction" / "executing an instruction"

Nazomi's Proposed Construction	Defendants' Proposed Construction
" instruction ": program commands interpreted or compiled into machine language	" instruction ": stack-based instruction
" executing an instruction ": Plain and ordinary meaning. No construction required.	" executing an instruction ": performing a stack-based instruction

The parties dispute whether an "instruction" must be "stack-based." Nazomi argues that requiring the instruction to be "stack-based" would improperly limit the claim based on embodiments in the specification. *See Howmedia Osteonics Corp. v. Wright Medical Tech., Inc.*, 540 F.3d 1337, 1345 (Fed. Cir. 2008). Nazomi contends that nothing in the plain language of the claim or specification indicates that an "instruction" is necessarily "stack-based."

Defendants counter that because claim 11 recites, "an instruction *that references an entry in a constant pool*," and because the constant pool is limited to Java, *see* Part III.A *infra* (concluding that "constant pool" was a term of art specific to Java at the time of invention), the instruction must necessarily be a stack-based Java instruction (or "bytecode"). Defendants also argue that a broader construction of "instruction" than theirs would not be enabled by the '160 patent specification. Defendants cite, *inter alia*, *Digital Biometrics, Inc. v. Identix, Inc.*, 149 F.3d 1335, 1344 (Fed. Cir. 1998) and *Athletic Alternatives, Inc. v. Prince Manufacturing, Inc.*, 73 F.3d 1537, 1581 (Fed. Cir. 1996) for the proposition that the court should adopt "a narrower, enabled construction over a broader non-enabled one." Defs.' Response Br. 12

Both parties point to the following passage in the specification as supporting their respective positions:

The term Java in the specification or claims should be construed to cover successor programming languages *or other programming languages using basic Java concepts* (the use of *generic instructions, such as bytecodes*, to indicate the operation of a virtual machine).

'160 Patent col.7 ll.35-39 (emphases added). Nazomi argues that "basic Java concepts" and "generic instructions, such as bytecodes" are not limited to stack-based Java instructions, while defendants argue that "basic Java concepts" and "generic instructions, such as bytecodes" are necessarily stack-based Java instructions.

This court previously construed the term "instruction" in the '215 parent patent to mean "*either a stack-based instruction that is to be translated into a register-based instruction, or a register-based instruction that is input to the CPU pipeline.*" *Nazomi Commc'ns, Inc. v. ARM Holdings, PLC*, 2006 WL 2578347, at *22 (N.D. Cal. 2006), *aff'd*, 266 Fed. Appx. 935 (Fed. Cir. 2008) ("*Nazomi 2002*") (emphases added). Common terms across patents that share the same disclosures are construed consistently across all claims in both patents. *Mycogen Plant Sci., Inc. v. Monsanto Co.*, 252 F.3d 1306, 1311 (Fed. Cir. 2001) (overruled on other grounds). The court construes the term "instruction" in the '160 patent consistently with the court's previous construction of the same term in the '215 parent patent in *Nazomi 2002* and concurrent construction in two related patents in Case No. C-10-04686, Dkt. No. 441.

In *Nazomi 2002*, this court held in the context of this patent family that, prior to translation, generic instructions are stack-based "Java bytecodes." 2006 WL 2578347, at *1, *6. It is not disputed that, in the JVM Specification, the instructions that reference the "constant pool" would be stack-based Java bytecodes. The court, however, rejects defendants' argument that because it construes "constant pool" as a Java-type constant pool attached to a single loaded class, it must necessarily limit the term "instruction" to stack based Java bytecodes. In contrast to the term "constant pool"—a term of art for which the court was presented *no definition* outside of the JVM Specification—here, the term "instruction" is broadly defined in the intrinsic record to include both generic, stack-based instructions ("Java bytecodes") and register-based instructions ("the native instructions to which those bytecodes are translated"). *Nazomi 2002*, 2006 WL 2578347, at *6, *22. In the '215 parent patent, the term "instruction" in the disputed claims was always preceded by the modifier "stack-based" or "register-based." In contrast, here, the unmodified term "instruction" in claim 11 presumably encompasses both types of instructions. *See, e.g., Seachange*, 413 F.3d at 1368-69. The '160 patent specification does not limit the instructions that reference the constant pool of the "present invention" to bytecodes. *See* '160 Patent col.2, ll.21-22; col.7 ll.58-61.

Moreover—although not briefed by the parties until their summary judgement briefs—claim 14, which depends on claim 11, recites the additional limitation that the "instruction comprises a bytecode." '160 patent col.10 ll.22-23. Because claim 14 is expressly limited to stack-based Java bytecodes, *see Nazomi 2002*, 2006 WL 2578347, at *6 (holding that bytecodes are stack-based Java bytecodes in the context of this patent family), it is presumed that claim 11 is not so limited. *See InterDigital Commc'ns, LLC v. Int'l Trade Comm'n*, 690 F.3d 1318, 1324-25 (Fed. Cir. 2012) ("[T]he presumption [of claim differentiation] is 'especially strong' in this case, because 'the limitation in dispute is the only meaningful difference between an independent and dependent claim, and one party is urging that the limitation in the dependent claim should be read into the dependent claim.'" (citing *Sunrace Roots Enterprise Co. v. SRAM Corp.*, 336 F.3d 1298, 1303 (Fed. Cir. 2003))). Defendants offer no convincing argument to overcome the presumption that claim differentiation applies here.

Defendants' assertion that the claimed process would only be enabled for stack-based instructions is based exclusively on attorney argument. A conclusion on enablement—a validity issue and not a claim construction issue—must be supported by clear and convincing evidence. 35 U.S.C. § 282; *Microsoft Corp. v. i4i Ltd. P'ship*, — U.S. —, 131 S. Ct. 2238, 2245 (2011); *cf.* *AquaTex Indus., Inc. v. Techniche Solutions*, 479 F.3d 1320, 1329 (Fed. Cir. 2007) (requiring "for example ' . . . testimony of experts or others versed in the technology . . . texts and treatises; . . . [or] disclosure in the prior art'" to prove infringement under the doctrine of equivalents, which only requires preponderant evidence (internal quotation omitted)). Here, defendants have not raised any invalidity challenge based on lack of enablement, and the court has insufficient evidence to reach any conclusion on this issue.

Moreover, the Federal Circuit has significantly cabined the doctrine stated in *Athletic Alternatives* (and applied in *Digital Biometrics*), i.e., the idea that a court should choose a claim construction that is enabled, to situations where "one patent applicant made two *contradictory* and *irreconcilable* affirmative representations of the contested limitation." *Housey Pharms., Inc. v. Astrazeneca UK Ltd.*, 366 F.3d 1348, 1356 (Fed. Cir. 2004) (emphasis in original); *Northern Telecom, Ltd. v. Samsung Elecs. Co.*, 215 F.3d 1281, 1295 (Fed. Cir. 2000) (limiting *Athletic Alternatives* to "the situation where two clear yet contradictory definitions are provided by the patentee"). The rule from *Athletic Alternatives* did not apply in *Housey* and *Northern Telecom* where there were "no contradictory and irreconcilable choices for the meaning of [the disputed term]." *Housey*, 366 F.3d at 1356; *see also Northern Telecom*, 215 F.3d at 1295 ("Under Samsung's reading, *Athletic Alternatives* would substitute for reasoned analysis. To the contrary, *Athletic Alternatives* considers the case where reasoned analysis leads to two clear and distinct definitions of claim language. It does not apply here, where confusing statements in the prosecution history simply fail to overcome the ordinary meaning of the [disputed] limitation."). Here, like in *Housey* and *Northern Telecom*, there is no indication that Nazomi provided "two contradictory and irreconcilable definitions" for the term "instruction." Reasoned analysis leads to one definition for "instruction" that is consistent with the claim, specification, and intrinsic record.

For these reasons, the court adopts the same construction as it did in *Nazomi 2002* and

concurrently adopts in Case No. C-10-04686. The court construes "instruction" to mean: "either a stack-based instruction that is to be translated into a register-based instruction, or a register-based instruction that is input to the CPU pipeline." Likewise, the court construes "executing an instruction" to mean "executing an instruction [as construed above]."

C. "an indication of a reference that may need resolution"

Nazomi's Proposed Construction	Defendants' Proposed Construction
data that indicates whether a reference has been resolved	an identification of a location within the constant pool that stores the name of a reference that may need resolution

The parties primarily dispute whether an "indication of a reference that may need resolution" indicates "a location within the constant pool that stores the name of a reference" (defendants' construction), or merely indicates "*whether* a reference has been resolved" (Nazomi's construction). Nazomi argues that the term's plain meaning supports its proposed construction. Nazomi also advances a claim differentiation argument. Claim 17 of the '160 patent recites: "[t]he method of claim 11, wherein the indication of a reference points to a *label* of an object that may or may not be loaded into memory." '160 Patent col.10 ll.28-30 (emphasis added). Nazomi argues that this court should not construe claim 11 and dependent claim 17 in a manner that renders the two claims equivalent in scope. *See Sunrace*, 336 F.3d at 1303.

Defendants counter that the "indication of a reference" must point to "the *name* of a reference" because "[t]he 'name' of the reference is the only information Nazomi disclosed in the specification that identifies or indicates a reference." Defs.' Response Br. 18-19. Defendants cite the following passage in the specification, describing figure 9A, for support: "The constant pool entries include a resolution data field 160 and indication field 162. In this case, the *indication '5002' points to address '5002' that contains the data 'ABC'* [where 'ABC' is the 'name.']" '160 Patent col.8 ll.24-27 (emphasis added). According to defendants, "the phrase 'identification of a location' [in their proposed construction] is broader than the phrase in claim 17 'points to a label.'" Defs.' Response Br. 20.

Nazomi's proposed claim construction—"data that indicates *whether a reference has been resolved*"—ignores the plain language of the claim. The claim requires "an indication of *a*

reference." *Id.* col.10 l.3 (emphasis added). Nazomi conflates the "indication of a reference that may need resolution" with the "data resolution field." According to Nazomi, both claim limitations identify *whether* a resolution has been resolved. But, the patent is clear that only the "resolution data field" indicates *whether* the reference has been resolved, for example, with an entry of "0," "1," "00," or "01." *Id.* col.10 ll.5-8 (claim 11: "using data from the resolution data field to determine *whether to do a resolving step* (emphasis added); col.8 ll.24-47 (Before resolution, "[t]he resolution data field indicates that the reference to the object has not been resolved.") and col.8 ll.7-13 (After resolution: "[t]he resolution data field is also set, so as to indicate that the reference is resolved."); figs.9A, 9B, and 10 (depicting the resolution data field as containing an entry of "0" or "1" (fig.9), or "00" or "01" (fig.10)). Similarly, the specification is clear that the "indication of a reference that may need resolution" is an indication, such as an address, that directs the system to a location within the constant pool that stores the name of a reference that needs resolution. With respect to figure 9A, the '160 patent specification describes the "indication of a reference" as follows: "the indication '5002' points to *address* '5002' that contains the data 'ABC.'" '160 patent col.8 ll.24-27 (emphasis added). With respect to figure 10, the specification describes a separate field "for the Java object *address* or nonJava *address*." *Id.* col.8 ll.53-54 (emphases added). Thus, the claim language and specification are explicit that the "resolution data field," and *not* the "indication of a reference that may need resolution," indicates *whether* a reference has been resolved.

Defendants' proposed construction is consistent with the plain language of the claim in view of the disclosures in the specification. The claim requires "an instruction . . . [that] references an entry in a constant pool, the constant pool entry storing an indication of a reference that may need resolution." *Id.* col.10 ll.2-4. The specification explains that the indication of a reference is a form of identification, such as an address, that directs the system to a location *within the constant pool* that stores the name, or "label," of the reference to be resolved. *See e.g., id.* col.8 ll.24-27 ("[T]he indication '5002' points to address '5002' that contains the data 'ABC.'"). Figure 10 depicts a constant pool entry as 32-bits. *See id.* fig.10 (allotting 2 bits for the "reference resolution field," 28 bits for the address, and 2 bits for "garbage collection"). Figure 10 simply shows that, in a 32 bit constant pool entry, the "indication field" would be 28 bits. *See id.* Figure 10, therefore, does not

1 provide any basis for adopting a broader definition for "indication of a reference," which is clearly
2 defined in the specification with respect to figure 9A. While the court understands that
3 embodiments in the specification are not generally limiting, *see Enzo Biochem, Inc. v. Applera*
4 *Corp.*, 599 F.3d 1325, 1342 (Fed. Cir. 2010) ("[I]t is improper to read limitations from a preferred
5 embodiment described in the specification—even if it is the only embodiment—into the claims
6 absent a clear indication in the intrinsic record that the patentee intended the claims to be so
7 limited."), here, the intrinsic record is clear that the embodiments disclosed in figures 8-10 *are* the
8 invention that is *claimed*. *See* '160 patent col.3 ll.16-17 ("*FIGS 8-10 illustrate the operation of the*
9 *present invention*") (emphasis added); *id.* col.7 l.61 ("[T]he *present invention of FIGS. 8-10* can be
10 completely implemented in software." (emphasis added)). Nazomi's proposed construction would
11 render the claims broader than that the disclosed invention. *Phillips*, 415 F.3d at 1316. As
12 discussed *supra*, Nazomi's alternative proposal conflates the "indication of a reference" with the
13 "resolution data field." Once the this theory is rejected, Nazomi does not present any argument in
14 the alternative that the "indication of a reference" would be anything other than what is disclosed in
15 the patent.

16 Although the court agrees with Nazomi that defendants' proposed construction is similar in
17 scope to dependent claim 17, the "presumption [of claim differentiation] is not a hard and fast rule
18 and will be overcome by a contrary construction dictated by the written description or prosecution
19 history." *Seachange*, 413 F.3d at 1369. Claims "written in different words may ultimately cover
20 substantially the same subject matter." *Id.* (citing *Multiform Desiccants*, 133 F.3d at 1480). Where,
21 as here, there is a conflict between the teachings of the specification and the doctrine of claim
22 differentiation, the teachings of the specification control. *See id.*

23 For these reasons, the court adopts defendants' construction with a slight modification,
24 construing the term "an indication of a reference that may need resolution" to mean: "an
25 identification of a location (e.g., an address) within the constant pool that stores the name, or 'label,'
26
27
28

of a reference that needs resolution."⁴ The court includes the word "label" for consistency with dependent claim 17.

D. "resolution data field" / "data resolution field"⁵

Nazomi's Proposed Construction	Defendants' Proposed Construction
location within the constant pool entry that stores an indication as to whether a reference has been resolved	a field within the constant pool that indicates whether a reference has been resolved, which is separate from the indication of the reference

The parties dispute whether the "resolution data field" must be separate from the indication of the reference that may need resolution. According to Nazomi, "[a] plain reading of the claims as understood by one of ordinary skill in the art would suggest that the 'indication of a reference that may need resolution' could be contained *in* the resolution data field." Pl.'s Br. 15-16. Nazomi's argument with respect to this claim limitation is based on the premise, rejected by the court, that the "indication of a reference that may need resolution" indicates *whether* a reference has been resolved. If this were true, Nazomi contends, the "resolution data field" would contain an "indication of a reference that may need resolution." Nazomi asserts that figure 10 depicts an embodiment wherein the "reference resolution field" indicates whether a reference has been resolved.

Defendants counter that the claims call for a separate resolution data field and indication of a reference that may need resolution. Defendants cite figures 9A, 9B, and 10 of the '160 patent, and the accompanying disclosures for support, which describe the "resolution data field 160" and "indication field 162" (figs.9A and 9B) and "reference resolution field 180a" and "java object address or non-java address [field] 180b" (fig.10), as separate fields within the constant pool entry.

As discussed in Part III.C *supra*, the "resolution data field" and the "indication of a reference that may need resolution" are distinct features of the invention. Claim 11 requires that the constant pool entry both: (1) "stor[e] an indication of a reference that may need resolution," and (2) "includ[e]

⁴ Once the reference is *resolved*, the indication field is updated to indicate the address of the resolved object *in memory*, and the field no longer stores "an indication of a reference that may need resolution."

⁵ The parties agree that these two terms have the same meaning in the claims at issue. Thus, the term "resolution data field," as used in this Order, includes both terms.

1 a resolution data field." '160 patent col.10 ll.3-4, 7. Figures 9A, 9B, and 10 all represent
2 embodiments wherein the "indication of a reference" is contained in a separate field from the
3 resolution data field (or "reference resolution field" in figure 10). As explained, the "resolution data
4 field" indicates *whether* the reference has been resolved, for example, with an entry of "0," "1," "00,"
5 or "01," and the "indication of a reference that may need resolution" directs the system to the name
6 or label of a particular reference that may need resolution. *See* Part III.C *supra*. There is no basis in
7 the intrinsic record to support Nazomi's argument that the 2-bit entry in field 180a in figure 10
8 (either "00" (unresolved java object), "01" (resolved java object), "10" (numeric constant), or "11"
9 (text string)), indicates *a reference* that may need resolution under the court's construction of that
10 term. The intrinsic evidence is clear that the data resolution field and the "indication of a reference
11 that may need resolution" are distinct requirements of the invention. *See Verizon Services v.*
12 *Vonage Holdings*, 503 F.3d 1295, 1308 (Fed. Cir. 2007) (limiting the scope of the patent to that
13 which is disclosed as "the invention" in the specification); *C.R. Bard, Inc. v. U.S. Surgical Corp.*,
14 388 F.3d 858, 863-66 (Fed. Cir. 2004) (holding that the summary of the invention and the rest of the
15 specification unequivocally defined the patent more narrowly than the claim language otherwise
16 indicated).

17 Although the court agrees with defendants that these two features of the invention must *both*
18 be present, the physical separateness of the resolution data field and the "indication of a reference
19 that may need resolution" is not properly addressed within the claim construction of this claim term.
20 The court's construction of "indication of a reference that may need resolution" disables Nazomi's
21 attempt to conflate the two claim limitations into one, *see* Part III.C *supra*, and thus renders
22 defendants' proposed physical separation requirement with respect to this term superfluous. The
23 court finds that the more appropriate approach to construing this term is to simply clarify that the
24 data in the data resolution field simply indicates *whether* a reference has been resolved, and does not
25 also "indicate a reference that may need resolution."

26 Accordingly, the court adopts the following construction for "resolution data field": "a data
27 field within the constant pool entry that contains data indicating whether a reference has been
28 resolved." Said data does not indicate a reference that may need resolution, as construed *supra*.

IV. ORDER

For the foregoing reasons, the court construes the disputed claim terms as follows:

Disputed Term	Construction
"constant pool"	a data structure attached to a single loaded class that encodes all the names that can be used by any method in the loaded class
"constant pool entry" / "entry in a constant pool"	an entry within the constant pool [as defined above]
"instruction"	either a stack-based instruction that is to be translated into a register-based instruction, or a register-based instruction that is input to the CPU pipeline
"executing an instruction"	executing an instruction [as defined above]
"an indication of a reference that may need resolution"	an identification of a location (e.g., an address) within the constant pool that stores the name, or "label," of a reference that needs resolution
"resolution data field" / "data resolution field"	a data field within the constant pool entry that contains data indicating whether a reference has been resolved

DATED: August 8, 2013


RONALD M. WHYTE
 United States District Judge